

Silly Rubber: An Implicit Material Point Method for Simulating Non-equilibrated Viscoelastic and Elastoplastic Solids

YU FANG, University of Pennsylvania

MINCHEN LI, University of Pennsylvania

MING GAO, University of Pennsylvania

CHENFANFU JIANG, University of Pennsylvania



Fig. 1. Our method easily captures the compliant and slowly-recovering behavior of a viscoelastic solid silicone rubber (left), as well as controllable flowing and merging behavior of creamy dense fluid foam (right) in a unified framework.

Simulating viscoelastic polymers and polymeric fluids requires a robust and accurate capture of elasticity and viscosity. The computation is known to become very challenging under large deformations and high viscosity. Drawing inspirations from return mapping based elastoplasticity treatment for granular materials, we present a finite strain integration scheme for general viscoelastic solids under arbitrarily large deformation and non-equilibrated flow. Our scheme is based on a predictor-corrector exponential mapping scheme on the principal strains from the deformation gradient, which closely resembles the conventional treatment for elastoplasticity and allows straightforward implementation into any existing constitutive models. We develop a new Material Point Method that is fully implicit on both elasticity and inelasticity using augmented Lagrangian optimization with various preconditioning strategies for highly efficient time integration. Our method not only handles viscoelasticity but also supports existing elastoplastic models including Drucker-Prager and von-Mises in a unified manner. We demonstrate the efficacy of our framework on various examples showing intricate and characteristic inelastic dynamics with competitive performance.

CCS Concepts: • **Computing methodologies** → **Physical simulation**.

Additional Key Words and Phrases: Material Point Method (MPM), augmented Lagrangian, viscoelasticity, elastoplasticity

Authors' addresses: Yu Fang, University of Pennsylvania; Minchen Li, University of Pennsylvania; Ming Gao, University of Pennsylvania; Chenfanfu Jiang, University of Pennsylvania.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM. 0730-0301/2019/7-ART118 \$15.00

<https://doi.org/10.1145/3306346.3322968>

ACM Reference Format:

Yu Fang, Minchen Li, Ming Gao, and Chenfanfu Jiang. 2019. Silly Rubber: An Implicit Material Point Method for Simulating Non-equilibrated Viscoelastic and Elastoplastic Solids. *ACM Trans. Graph.* 38, 4, Article 118 (July 2019), 13 pages. <https://doi.org/10.1145/3306346.3322968>

1 INTRODUCTION

Many real-world materials and natural phenomena involve inelastic deformation. Common examples include metal bending under pressure, sand flowing down a slope, and whipped cream forming a captivating shape on a cake. In computer graphics, since the pioneering work of Terzopoulos et al. [1988; 1987], the study of viscoelasticity, plasticity, and fracture has attracted increasing attention from numerous researchers, not only due to the ubiquity of these behaviors but also due to the complex and intriguing dynamics they exhibit.

Recently, the Material Point Method (MPM) [Sulsky et al. 1995] has gained popularity for inelastic materials due to its flexible geometric representation using meshless particles. Since inelastic objects tend to undergo extreme deformation with permanent shape change from their rest configurations, representing them with meshes as in the Finite Element Method (FEM) imposes the additional difficulties of mesh distortion and tangling. MPM, on the other hand, uses particles to track the history of all strain and stress states and relies on a background grid to accurately evaluate derivatives in force computations. The superior treatment of extreme deformations and topology changes has enabled MPM to be used for simulating many inelastic phenomena in computer graphics, showcased by the successful simulation of materials such as snow [Gaume et al. 2018; Stomakhin et al. 2013], sand [Daviet and Bertails-Descoubes 2016; Klár et al. 2016; Yue et al. 2018], sponge and foam [Ram et al. 2015], and non-Newtonian fluids [Yue et al. 2015].

Compared to the prevalent use of finite strain plasticity in granular materials such as snow and sand [Stomakhin et al. 2013; Yue et al. 2018], viscoelastic models are less developed for MPM in the literature, despite its apparent applications in simulating a wide variety of materials ranging from solid-like objects (such as soft biological tissues, polymeric foams and silicone rubbers) to fluid-like objects (such as whipped cream). The latter was investigated by Ram et al. [2015] using the Oldroyd-B constitutive model for describing viscoelastic shear-thinning fluids. Their formulation relies on the discretization of the upper convected derivative of the right Cauchy Green strain [Bonet and Wood 2008] and requires considerable additional effort to combine with standard existing MPM schemes formulated using the deformation gradient evolution [Stomakhin et al. 2013]. Furthermore, the viscoelastic effect is fully lagged in their formulation, resulting in a semi-implicit integration scheme that may overestimate the elastic effect [Pradhana et al. 2017]. Around the same time, Yue et al. [2015] also simulated similar Non-Newtonian fluid effects using a viscoplastic Bingham model with a focus on dense foams.

Even though viscoplasticity and viscoelasticity are both continuum mechanics models for describing rate-dependent effects and have both been used for simulating foam and cream like materials, their underlying principles are very different. Since we focus on materials such as rubber polymers and solid foams with large elastic deformation, we assume *finite strain elasticity* and ignore the incompressible limit throughout the paper. Within this regime, unlike Bingham plastic fluids, visco(elasto)plasticity assumes the existence of yield stress like plasticity, and the material deforms elastically without any rate-dependent plastic flow before yielding. In contrast, viscoelasticity describes thermodynamic dissipation from non-equilibrated (time-dependent) elastic deformation, which always exists regardless of the severity of deformation as long as the continuum is not at equilibrium. As the term itself suggests, viscoelasticity utilizes elasticity for resisting large deformations and viscosity for resisting rapid change of deformations.

Critical applications we focus on include both the behaviors of solid materials like viscoelastic silicone rubber and solid foams (such as mattresses), and the flowing and creamy behaviors of viscoelastic fluids like dense liquid foams and melted cheese.

Another similar ingredient in physics-based simulation for dissipative results is damping. Note that damping, such as the commonly used Rayleigh damping in graphics, is essentially a response filter that damps out high-frequency vibrations during elastic deformations. In this paper, we introduce an MPM transfer-based damping scheme as well, and our viscoelasticity treatment should not be mistakenly treated as a damping scheme due to its principled physical basis on non-equilibrated dynamics. In Table 1 we compare various properties of inelastic models and damping.

We target the notable lack of an easy-to-implement viscoelasticity treatment for MPM, and present a hybrid particle/grid numerical framework for finite strain viscoelasticity based on decomposing the deformation gradient into a viscous part and an elastic non-equilibrated part. The resulting formulation for the modification of the non-equilibrated elastic strain can be implemented in a predictor-corrector algorithm that is remarkably similar to the exponential

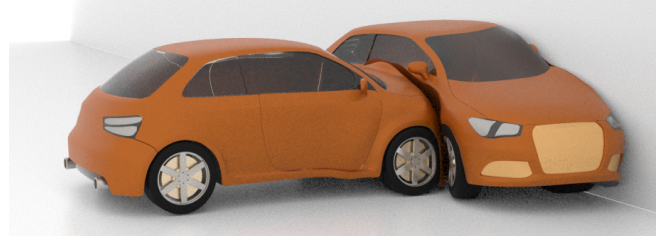


Fig. 2. **Car crash.** Our ADMM solver handles von-Mises plasticity well as demonstrated in this car crash scene. The cars quickly end up fully stopped due to substantial plastic dissipation.

return mapping algorithm used for finite strain plasticity, thus enabling a straightforward implementation into any existing MPM frameworks that already support plasticity.

For numerical integration, both explicit and implicit MPM schemes are well suited for our model. For stiff or highly viscous materials, implicit integration is preferred to avoid small time steps. While Newton’s method has historically been the standard nonlinear implicit solver for the Finite Element Method in computer graphics [Sifakis and Barbić 2015], recent advances in the optimization-based reformulation of backward Euler hyperelasticity [Gast et al. 2015] bring alternative schemes to Newton’s method, such as Projective Dynamics [Bouaziz et al. 2014], Alternating Direction Methods of Multipliers (ADMM) [Overby et al. 2017] and Quasi-Newton methods with L-BFGS [Liu et al. 2017]. Inspired by the ADMM formulation for FEM hyperelasticity, we formulate our implicit MPM inelasticity problem into a pseudo-optimization problem. Here we use the term “pseudo-optimization” because we cannot explicitly express the objective function as an energy. Regardless, we show that the ideas of augmented Lagrangian and ADMM can still be applied to our formulation, allowing efficient simulation of elastoplastic and viscoelastic solids with backward Euler integration that avoids the expensive solves associated with nonsymmetric global linear systems as in previous MPM works based on Newton’s method [Klár et al. 2016].

In summary, the main contributions of our work are:

- An easy-to-implement non-equilibrated viscoelasticity treatment for MPM that has a similar structure to the exponential return mapping algorithm for elastoplasticity.
- The formulation of finite strain implicit elastoplasticity and viscoelasticity into a pseudo-optimization problem that allows an augmented Lagrangian treatment with operator splitting.
- The extension of the ADMM solver into MPM, supporting the simulation of common inelastic materials.

Table 1. Comparing properties of inelasticity and damping models.

	Elastoplasticity	Viscoplasticity	Viscoelasticity	Damping
Change of rest state	✓	✓	✗ ^a	✗
Existence of yield stress	✓	✓	✗	✗
Deformation rate dependent	✗	✓	✓	✗
Dissipative	✓	✓	✓	✓

^aViscoelastic solids have similar rest shape to elastic ones; some viscoelastic fluids (ignoring the incompressible limit) only have rest density.



Fig. 3. **Warrior**. A long and thick viscoelastic sheet drops onto a rotating statue, producing complex folding behaviors.

- A numerical damping strategy based on the Rigid Particle-In-Cell (RPIC) method [Jiang et al. 2015] that is unconditionally stable and has equivalence to Newtonian viscosity.

We also explore several strategies to improve the convergence of the optimization-based inelasticity solver in §7.3. We demonstrate the efficacy of our method on materials ranging from compliant and shape-recovering viscoelastic solids (Fig. 1, 5), to buckling and coiling viscoelastic fluids (Fig. 1, 3), to yielding and flowing plastic media (Fig. 7, 16, 2) with characteristic visual appearances and competitive performance.

2 RELATED WORK

Inelasticity. Elastic models have been extensively studied in graphics. With similar popularity, inelastic materials like foam (which undergo rate-dependent or permanent distortion under external loads) also attract much attention [Terzopoulos and Fleischer 1988]. Different discretization methods have been proposed to model elastoplasticity, including mesh-based methods [Wicke et al. 2010], particle-based methods [Chentanez et al. 2016; Gerszewski et al. 2009; He et al. 2018; Jones et al. 2014] and hybrid methods [Gao et al. 2017; Gaume et al. 2018; Jiang et al. 2017; Klár et al. 2016].

Viscoelastic materials exhibit both viscous and elastic behaviors under external loads. Simple models (e.g. the Maxwell model) can be achieved by combining an elastic spring and a viscous damper as first studied by [Terzopoulos and Fleischer 1988]. For viscoplasticity, the deformation becomes unrecoverable when the loads reach yielding points. In recent years, more and more works seek to simulate viscoelasticity and viscoplasticity via two main strategies.

Works utilizing the first approach start from the Navier-Stokes equations and add additional terms corresponding to elastic and viscous forces to an incompressible liquid solver [Carlson et al. 2002; Goktekin et al. 2004; Losasso et al. 2006; Rasmussen et al. 2004; Takahashi et al. 2015, 2014]. These methods generate visually appealing results but usually require performing an incompressibility projection for fluid motion, and it becomes more expensive when implicit viscosity needs to be enforced with a separate linear solve. Batty et al. [2008; 2011; 2012] extend this to thin sheets and accelerate the computation by adaptively remeshing the background meshes. Larionov et al. [2017] further improve the accuracy and realism by solving a unified pressure and viscosity system. Barreiro et al.

[2017] employ a conformation tensor to capture a wide range of viscoelastic fluid behaviors. Zhu et al. [2015] simulate non-Newtonian viscoplastic flow on flexible meshes.

The second direction is rooted in finite strain solid mechanics: constitutive models are designed to relate the deformation strains to the reactive forces. Early works include mesh-based Finite Element methods [Bargteil et al. 2007; Wojtan and Turk 2008] which may require frequent remeshing to avoid ill-conditioned system solves under significant distortion. More recently, the hybrid particle-grid MPM was demonstrated to be capable of simulating some viscoelastic and viscoplastic materials [Ram et al. 2015; Yue et al. 2015] without geometric difficulties. We build our method with MPM due to this advantage.

Compared to most prior work, our method more easily captures solid foam behavior under large deformation, including compliance to gentle deformation and slow *recovery* to a memorized rest shape. Examples include filled rubbers, polymers, and polymeric foams (everyday objects like vehicle tires, sofas, mattresses, and pillows). Additional major features of our model are (1) the unified treatment of viscoelasticity and elastoplasticity with finite strain, (2) a unified constitutive model for controlling fluid (cream-like) and solid (solid foam-like) behaviors, (3) an efficient implicit solver, and (4) an intrinsic ability to couple with other materials due to the MPM framework.

Optimization for Elasticity. Many equations in simulation can be reformulated into equivalent optimization problems, enabling more efficient and robust solvers. Teran et al. [2005] modify the Hessian of the optimization function to enforce positive definiteness of the global stiffness matrix. Batty et al. [2007] propose a variational interpretation of the traditional pressure projection in fluid simulations by minimizing the kinetic energy under incompressibility constraints. For dynamic elasticity, a balance between inertia and the elastic potential needs to be computed instead. Due to the stringent performance requirements, accuracy could be traded for efficiency in real-time applications. Some recent works [Gast et al. 2015; Liu et al. 2017] focus more on second-order Newton or Quasi-Newton methods. Another trend in the literature is based on first-order methods. Wang and Yin [Wang and Yang 2016] propose a gradient descent solver on GPU platforms. Local-global style solvers [Bouaziz et al. 2014; Liu et al. 2013] alternate between small local solves and a large but simple global solve. Earlier works are

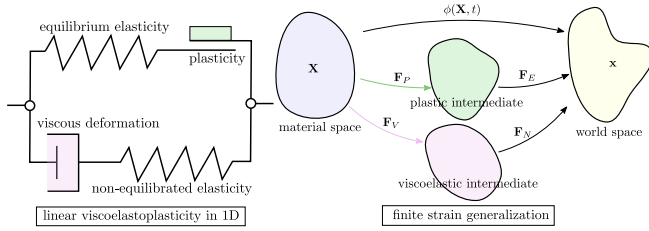


Fig. 4. The underlying viscoelastoplasticity model: 1D rheological model and 3D finite strain multiplicative decomposition of the deformation gradient.

restricted to quadratic energies. Overby et al. [2017] accommodate arbitrary isotropic nonlinear elasticity using ADMM. Peng et al. [2018] reinterpret the local-global steps as a fixed-point problem; thus Anderson acceleration can be applied to accelerate the convergence. We also introduce several practical strategies to improve the convergence of our ADMM MPM inelasticity solver in §7.3.

As a natural extension, damping has been integrated into optimization based elasticity integrators as in [Brown et al. 2018; Dinev et al. 2018a,b; Gast et al. 2015; Li et al. 2018]. While some of them are easy to implement, they may suffer from undesired artifacts such as how Ether drag does not conserve linear momentum. On the other hand, sophisticated alternatives could potentially require intensive modifications to existing pipelines. Furthermore, most methods cannot be interchangeably applied to both explicit and implicit time integrations. In this work, we present an unconditionally stable and straightforward damping method for MPM which is integration scheme invariant and momentum and angular momentum preserving.

In this paper, we adopt an Augmented Lagrangian formulation for the simulation of inelasticity within MPM. In the computational fluid mechanics literature, Augmented Lagrangian methods using similar variable splitting schemes have been applied to simulating viscoplastic [Fortin and Glowinski 1983] and viscoelastic [Fortin and Fortin 1989] flows. Generalized standard materials defined by an elastic energy density and a dissipation potential also provide another possibility for the energetic formulation of viscoelastic materials [Le Tallec 1990; Mielke 2006].

Material Point Method. As a hybrid method that generalizes FLIP [Zhu and Bridson 2005], MPM has become popular recently in the graphics community due to its efficiency for simulating animations with large deformations and large topology changes. It has proven capable of capturing a wide breadth of natural phenomena, including both single-species materials, e.g., foam [Ram et al. 2015; Yue et al. 2015], snow [Gaume et al. 2018; Stomakhin et al. 2013], sand [Daviet and Bertails-Descoubes 2016; Klár et al. 2016; Yue et al. 2018], cloth [Guo et al. 2018; Jiang et al. 2017], fracture [Wretborn et al. 2017], and multi-species multi-phases mixtures, e.g., fluid-solid mixture [Fei et al. 2018; Gao et al. 2018a,b; Pradhana et al. 2017; Stomakhin et al. 2014; Yan et al. 2018].

3 CONSTITUTIVE MODEL

Finite strain elastoplasticity has been widely adopted in much of computer graphics research including both mesh-based FEM simulations [Irving et al. 2004; Wojtan and Turk 2008] and MPM simulations [Klár et al. 2016; Yue et al. 2018]. We refer to Simo’s article [1992] and book [Simo and Hughes 2006] for a more detailed explanation of the core theory. While viscoelastic *fluids* have been largely studied in computer graphics [Barreiro et al. 2017; Batty et al. 2012; Ram et al. 2015], an alternative formulation with finite strains for *silicone rubber-like viscoelastic solids* with plasticity-like simple treatment has not been well described beyond the *viscoplasticity* work of Bargteil and Wojtan et al. [2007; 2008]. In this section, we present a highly flexible and easy-to-implement constitutive model for finite strain viscoelasticity that supports large deformation.

3.1 Kinematics and Motivation

Our treatment for viscoelasticity is built upon the theory of Nedjar [2002a] which accounts for the combination of viscoelasticity and plasticity for large deformations. Note that to prevent non-physical stress during rotation, a *finite strain* treatment is necessary for scenes with large elastic deformation (such as silicone rubber). For mostly fluid-like materials (toothpaste, liquid foam, cream etc.), adopting a more straightforward linear model as in Goktekin et al.’s work [2004] is also a favorable choice.

Let us assume a general viscoelastic material that also potentially (but not necessarily) undergoes elastoplastic yielding. In the large deformation regime, the deformation map from the material space Ω^0 (with coordinate $\mathbf{X}$) to the world space Ω^n (with coordinate $\mathbf{x}$) is usually described as $\mathbf{x} = \phi(\mathbf{X}, t)$. As illustrated with the 1D rheological model in Fig. 4 (left), the simulated solid consists of a parallel combination of (1) an elastic spring with a frictional element for plasticity and (2) another spring with a viscous dash-pot element which is assumed to capture the dissipation due to thermodynamic non-equilibrium. When extending to 3D nonlinear material responses, we characterize the generalization of this model with the multiplicative decomposition of the deformation gradient $\mathbf{F} = \frac{\partial \phi}{\partial \mathbf{X}}(\mathbf{X}, t)$. As demonstrated in Fig. 4 (right), $\mathbf{F}$ can be decomposed in two ways [Govindjee and Reese 1997]:

$$\mathbf{F} = \mathbf{F}_E \mathbf{F}_P = \mathbf{F}_N \mathbf{F}_V, \quad (1)$$

where $\mathbf{F}_E$ is the elastic deformation gradient at thermodynamic equilibrium, $\mathbf{F}_P$ is the plastic deformation gradient, $\mathbf{F}_N$ is the non-equilibrated elastic deformation gradient, and $\mathbf{F}_V$ is the viscous deformation gradient. Its 1D analogy directly motivates such a decomposition:

- (1) $\mathbf{F}_P$ characterizes the response of plastic or frictional elements, and $\mathbf{F}_E$ is the elastic deformation when the whole system is at equilibrium. This part of the model is time-independent.
- (2) When the system is not at equilibrium, $\mathbf{F}_N$ denotes a time-dependent elastic response (analogous to the spring connected to the dashpot in 1D). $\mathbf{F}_V$, representing the dashpot, captures the viscous part of the deformation gradient due to its rate of change and does not contribute to the total elastic potential energy.

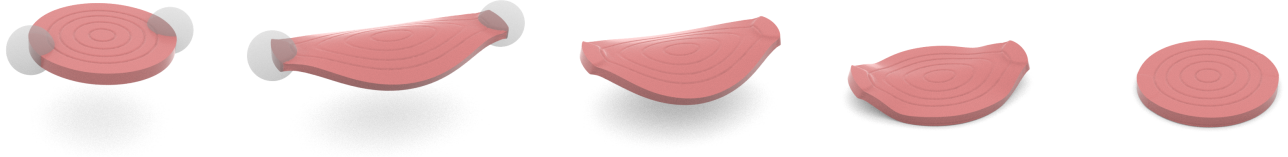


Fig. 5. **Viscoelastic silicone rubber** complies to slow deformation. After sufficient relaxation and releasing, it slowly deforms back to its original shape due to non-equilibrated viscosity.

Here the terms “equilibrated” and “non-equilibrated” parts correspond to time-independent and viscous (time-dependent) parts of the model. Note that for a purely viscoelastic solid without any plasticity, we have $F_P = I$ and $F = F_E = F_N F_V$. On the other hand, for a purely elastoplastic solid without any viscosity, it is typical to adopt $F = F_E F_P$ [Simo 1992].

Besides our multiplicative decomposition in Eq. (1), there also exist other popular visco-elasto-plastic models. For materials like rubber polymers and solid foams, another popular choice is to put an equilibrated spring in parallel with a dashpot-spring series; then they are connected to a sliding friction element as a whole. Generalization of this choice results in a decomposition of $F = F_E F_P$, $F_E = F_N F_V$, and thus $F = F_N F_V F_P$ in the finite strain case, which implies strong coupling of the two dissipative behaviors (plasticity and viscosity) and is harder to treat than our choice from an algorithmic perspective. We leave explorations along this direction in future work.

3.2 Mechanical Response

The elastoplastic part of the mechanical response can be treated as elastic stress τ_E that is only related to F_E . The plastic flow rule decides, for any rate of the total deformation gradient, how F_P (and thus F_E) should evolve in a thermodynamically consistent way. Due to the vast amount of previous work [Gao et al. 2017; Klár et al. 2016; Yue et al. 2015, 2018], we skip the detailed derivation of the plastic flow rule for common plastic materials and focus on the non-equilibrated viscoelastic part.

We will assume the simulated material of interest is isotropic. When viscoelastic responses exist, the total elastic potential energy density can be expressed as the direct summation of the contributions from an equilibrated part from F_E and a non-equilibrated part from F_N :

$$\Psi(F_E, F_N) = \Psi_E(F_E) + \Psi_N(F_N). \quad (2)$$

Note that both Ψ_E and Ψ_N can take the form of standard hyperelastic energy densities. In this paper, we use the isotropic St. Venant Kirchhoff model with the Hencky strain measure [Klár et al. 2016] since it gives a linear relationship between the Kirchhoff stress and the Hencky strain.

With finite viscoelasticity, a creep-like potential of dissipation $\Phi_V(\tau_N)$ is usually assumed to exist [Reese and Govindjee 1998] (where τ_N is the Kirchhoff stress tensor associated with F_N through some constitutive relationship), such that the rate of the non-equilibrated

right Cauchy-Green elastic strain $b_N = F_N F_N^T$ can be written as

$$\frac{Db_N}{Dt} = (\nabla \mathbf{v})b_N + b_N(\nabla \mathbf{v}) + \mathcal{L}_{\mathbf{v}}b_N, \quad (3)$$

where $\mathcal{L}_{\mathbf{v}}b_N$ is the Lie derivative of b_N with respect to velocity $\mathbf{v}$, and is given by [Nedjar 2002a] $\mathcal{L}_{\mathbf{v}}b_N = -2 \frac{\partial \Phi_V}{\partial \tau_N}(\tau_N)b_N$. The dissipation potential $\Phi_V(\tau_N)$ is designed to ensure that the mechanical work done by the viscous stress is always dissipative.

4 ALGORITHM OVERVIEW

We describe the predictor-corrector scheme for inelasticity in §5. Subsequently, §6 incorporates this model into the standard MPM pipeline. For implicit integration, we describe an Augmented Lagrangian solver in §7. Finally, a useful damping scheme for MPM in general is presented in §8.

5 PREDICTOR-CORRECTOR MAPPING

In the discrete setting, enforcing inelasticity is in essence a matter of discretizing the elastic strain rate (e.g., Eq. (3)) in a numerically stable and physically plausible manner. As in elastoplasticity, Eq. (3) can be integrated in a predictor-corrector scheme, where the predictor step advances b_N from time t^n to a trial state b_N^{tr} assuming there is no viscosity, then the corrector step integrates the contribution of $\mathcal{L}_{\mathbf{v}}b_N$ to get the corrected strain at time t^{n+1} .

5.1 Elastic Predictor

Our formulation allows independent trial steps for the equilibrated and non-equilibrated elastic deformation gradients. Since the deformation rate can be expressed as $\frac{DF}{Dt} = (\nabla \mathbf{v})F$, the elastic predictor stages for elastoplasticity and viscoelasticity are

$$F_{\star}^{\text{tr}} = (I + \Delta t \nabla \mathbf{v})F_{\star}^n, \quad \star \in \{E, N\}. \quad (4)$$

Note that in MPM, the Eulerian velocity gradient $\nabla \mathbf{v}$ can be easily evaluated on a particle using the grid velocities, which allows for a highly efficient predictor stage independently performed on each particle.

5.2 Inelastic Corrector

The elastoplastic corrector is usually done through the return mapping algorithm [Klár et al. 2016; Yue et al. 2015]. For simplicity of notation, we express it as a projection on F_E^{tr} denoted with $F_E^{n+1} = \mathcal{Z}_E(F_E^{\text{tr}})$, where $\mathcal{Z}_E$ is usually a piecewise function that enforces the feasibility of the elastic stress based on the location of F_E^{tr} [Klár et al. 2016].

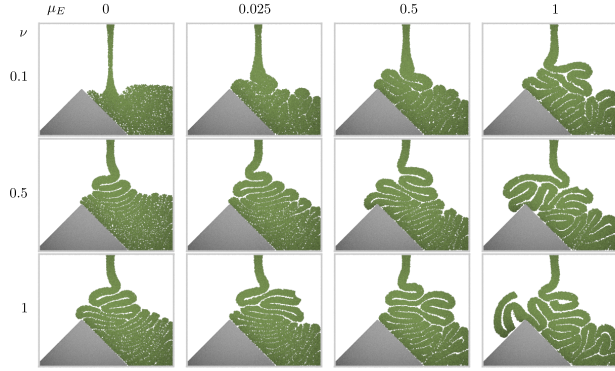


Fig. 6. **The effect of equilibrated shear modulus and viscosity.** μ_E controls the fluidity; when it is close to zero, the material has negligible shearing resistance to retain its shape and becomes a fluid. ν , the viscosity coefficient, acts as a viscous damper such that larger ν values cause more resistance to rapid deformation through non-equilibrated elasticity.

It turns out that the corrector for F_N^{tr} can also be explicitly written as a projection. In particular for isotropic materials, if we adopt the singular value decomposition $F_N = U_N \Sigma_N V_N^T$ and introduce the principal diagonal Hencky strain $\epsilon_N = \langle \log \Sigma_N \rangle$ where operator $\langle \cdot \rangle$ maps a diagonal matrix to its vector form, then the implicit discretization of $\frac{D b_N}{D t} = \mathcal{L}_\nu b_N$ can be shown to reduce to a potentially nonlinear equation system for ϵ^{n+1} [Nedjar 2002b]

$$\epsilon_N^{n+1} = \epsilon_N^{\text{tr}} - \Delta t \frac{\partial \Phi_V}{\partial \tau_N}(\tau_N^{n+1}(\epsilon_N^{n+1})), \quad (5)$$

where $\tau_N = \frac{\partial \Psi_N}{\partial \epsilon_N}$ is the principal Kirchhoff stress in vector form.

Non-equilibrated elasticity. For the non-equilibrated energy density Ψ_N , we adopt the St. Venant Kirchhoff model with the Hencky strain measure, which can be expressed in the principal strain space Σ_N as $\Psi_N(\Sigma_N) = \mu_N \text{tr}((\log \Sigma_N)^2) + \frac{1}{2} \lambda_N (\text{tr}(\log \Sigma_N))^2$, with the corresponding principal Kirchhoff stress

$$\tau_N = \frac{\partial \Psi_N}{\partial \epsilon_N} = 2\mu_N \epsilon_N + \lambda_N \text{tr}(\epsilon_N) \mathbf{1}, \quad (6)$$

where $\mathbf{1}$ denotes the all-ones vector, μ_N and λ_N are the material Lamé parameters.

Dissipation potential. The pseudo-potential of dissipation $\Phi_V(\tau_N)$ is chosen to be quadratic in terms of the principal non-equilibrated stress: $\Phi_V(\tau_N) = \frac{1}{2\nu_d} |\text{dev}(\tau_N)|^2 + \frac{1}{9\nu_v} (\tau_N \cdot \mathbf{1})^2$, where $\text{dev}(\tau_N) = \tau_N - \frac{1}{d} (\tau_N \cdot \mathbf{1}) \mathbf{1}$ is the deviatoric part of the principal stress and $d = 2$ or 3 is the problem dimension. ν_d and ν_v are viscosity parameters separately controlling the dissipation of the deviatoric and dilational parts of the non-equilibrated stress. The derivative of the dissipation potential is

$$\frac{\partial \Phi_V}{\partial \tau_N}(\tau_N) = \frac{1}{\nu_d} \text{dev}(\tau_N) + \frac{2}{9\nu_v} \text{tr}(\tau_N) \mathbf{1}, \quad (7)$$

which separately controls the viscous flow on the deviatoric and dilational non-equilibrated stresses.

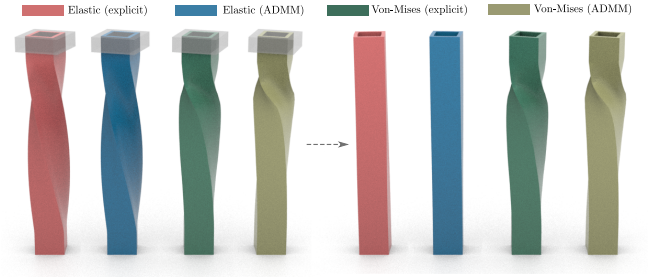


Fig. 7. **Twisting.** Our ADMM-based integrator reveals similar behaviors to explicit integration for hyperelastic and inelastic (von-Mises) models.

Correcting the strain. Substituting Eq. (6) and (7) into (5) gives $\epsilon_N^{n+1} = \epsilon_N^{\text{tr}} - \Delta t (\alpha \epsilon_N^{n+1} + \beta \text{tr}(\epsilon_N^{n+1}) \mathbf{1})$, where $\alpha = \frac{2\mu_N}{\nu_d}$ and $\beta = \frac{2(2\mu_N + \lambda_N d)}{9\nu_v} - \frac{2\mu_N}{\nu_d d}$ are constants. It has an analytic solution of

$$\epsilon_N^{n+1} = A (\epsilon_N^{\text{tr}} - B \text{tr}(\epsilon_N^{\text{tr}}) \mathbf{1}), \quad (8)$$

where $A = \frac{1}{1 + \Delta t \alpha}$, $B = \frac{\Delta t \beta}{1 + \Delta t (\alpha + d\beta)}$. Note that more complex nonlinear dissipation potentials for $\Phi_V(\tau_N)$ could also be chosen for more versatile dissipation behaviours, which would potentially cause the equations for ϵ_N^{n+1} to become a nonlinear system that requires Newton's method to solve. Since $\epsilon_N^{n+1} = \langle \log(\Sigma_N^{n+1}) \rangle$, Eq. (8) defines an analytic and smooth projection $F_N^{n+1} = \mathcal{Z}_N(F_N^{\text{tr}})$. Equivalently, the projection can be expressed in the principal space as $\Sigma_N^{n+1} = \hat{\mathcal{Z}}_N(\Sigma_N^{\text{tr}})$, whose derivative is given by

$$\frac{\partial \hat{\mathcal{Z}}_N}{\partial \Sigma_N}(\Sigma_N^{\text{tr}}) = \Sigma_N^{n+1} A \left((\Sigma_N^{\text{tr}})^{-1} - B \mathbf{1} \langle (\Sigma_N^{\text{tr}})^{-1} \rangle^T \right).$$

As we explain later, such a derivative plays an important part in our implicit inelasticity integration.

6 MATERIAL POINT METHOD

Our viscoelasticity scheme fits well into the standard MPM discretization scheme [Sulsky et al. 1995] due to (1) its similarity to plasticity return mapping, and (2) the embarrassingly independent strain treatment on MPM particles. MPM uses meshless particles to track the mass, volume, deformation gradients, and stress over the whole continuum, and an Eulerian background grid as a scratchpad for solving the governing equations in each time step. Throughout this work we use subscripts p, q for particle indices and i, j, k for background grid nodal indices. We use superscript $0, n, n+1$ for time discretization instances.

We assume the usage of Affine Particle-In-Cell [Jiang et al. 2015] particle/grid transfers for stability and Moving Least Squares MPM [Hu et al. 2018] force computation for efficiency. At time t^n , MPM typically lets particles carry mass m_p , original volume V_p^0 , position $\mathbf{x}_p^n$, velocity $\mathbf{v}_p^n$, affine velocity gradient C_p^n and elastic deformation gradient F_E^n . For viscoelasticity, we additionally store the non-equilibrated elastic deformation gradient F_N^n . The core steps from t^n to t^{n+1} are:

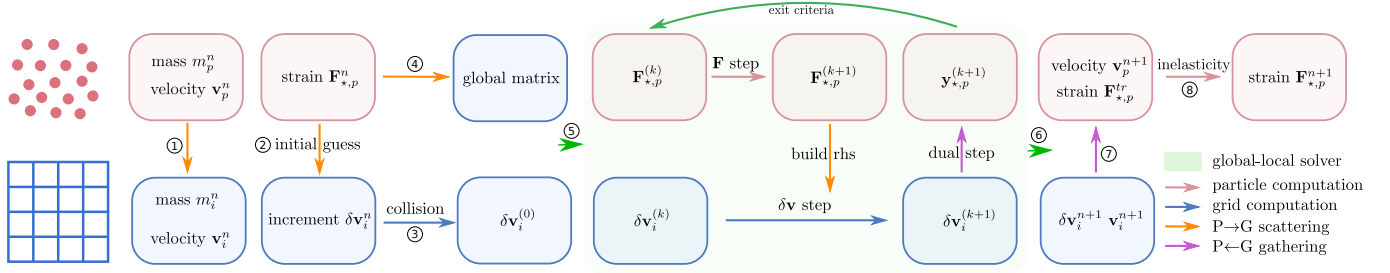


Fig. 8. **Algorithm flow.** In the pre-computation stage, we perform the standard P2G, evaluate and store the global matrix from $F_{\star}^n$, and also enforce that the initial guess of nodal velocity changes satisfies the boundary condition. Then, within the ADMM stage, we alternate particle step, grid step, and dual step sequentially. After the exit criterion is met, although we have both deformation gradients and nodal velocity changes available, only the latter one is adopted and then used to evolve the trial deformation gradient to ensure consistency in the G2P stage. Proper inelastic projection can be optionally applied afterward.

- (1) **Precomputation.** We sort particles based on their positions for better memory locality. Optional numerical damping can be added according to the method described in §8.
- (2) **Particles-to-grid.** Grid mass m_i^n and velocities $\mathbf{v}_i^n$ are transferred from particles using APIC [Jiang et al. 2015].
- (3) **Explicit force computation.** Explicit forces on the grid $\mathbf{f}_i^n$ are computed using MLS-MPM [Hu et al. 2018]. This contains contributions from gravity, equilibrated elastic stress $\frac{\partial \Psi_E}{\partial \mathbf{F}_E}(\mathbf{F}_E^n)$, and non-equilibrated elastic stress $\frac{\partial \Psi_N}{\partial \mathbf{F}_N}(\mathbf{F}_N^n)$.
- (4) **Evaluating velocity increment.** For symplectic Euler integration, the grid velocity increment is simply $\delta \mathbf{v}_i = \mathbf{f}_i^n \Delta t$. However, for a backward Euler discretization of implicit inelasticity, we need to solve a non-linear system of equations to get $\delta \mathbf{v}$ (see §7). Boundary conditions are enforced on $\delta \mathbf{v}$ (§7.2).
- (5) **Grid-to-particles.** By having $\mathbf{v}_i^{n+1} = \mathbf{v}_i^n + \delta \mathbf{v}_i$, APIC can be used again to get the updated $\mathbf{v}_p^{n+1}$ and $\mathbf{C}_p^{n+1}$.
- (6) **Particle strain update.** Strains $\mathbf{F}_E^{n+1}$ and $\mathbf{F}_N^{n+1}$ are updated through the corresponding prediction-correction scheme to account for plasticity and viscoelasticity effects (§5.2).
- (7) **Particle advection** with $\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \mathbf{v}_p^{n+1} \Delta t$.

It is clear that for an existing MPM elastoplasticity framework with explicit integration, only three additional steps (non-equilibrated force computation, strain prediction, and strain correction) are needed for adding viscoelasticity. Fully implicit integration is sometimes favored, especially for highly stiff materials (in which case, symplectic Euler suffers from extremely small time step sizes due to the existence of the CFL condition [Fang et al. 2018]). To handle such cases, we present a pseudo-optimization-based implicit solver in §7. We also give its algorithmic flow in Fig. 8.

7 AUGMENTED LAGRANGIAN SOLVER

To handle stiff materials with large time steps, implicit time integration schemes such as backward Euler are preferred, especially for large deformations [Gast et al. 2015]. However, it is more complicated when it comes to inelastic problems due to the lack of an analytic smooth energy that directly results in force balance. Klár et al. [2016] apply backward Euler to granular materials at the cost

of solving a non-symmetric linear system in each Newton iteration. Furthermore, a C^0 smooth plasticity projection operator ($\mathcal{Z}$) from Drucker-Prager leads to poor stability since line search cannot be used. As a result, Klár et al. [2016] have to take tiny time steps, resulting in an implicit solver that is usually slower than explicit alternatives. Considering these difficulties, we instead adopt an augmented Lagrangian solver enhanced with ADMM, a variable splitting solver, to reduce the difficulty associated with the non-smooth problem.

Overby et al. [2017] introduced ADMM to simulate real-time mesh-based elasticity with great success. ADMM alternates between solving two sets of primal variables and one additional set of dual variables that serves to iteratively adjust the “forces” applied to each set of the primal variables. We show that by including inelasticity into the formulation, ADMM successfully decomposes the high-dimensional non-smoothness associated with the large implicit system of the inelastic problem on the MPM grid into numerous low-dimensional non-smoothness within each particle’s local systems – allowing large time steps with stability and visually plausible behaviors.

7.1 Formulation

When considering inelastic problems, an optimization-based reformulation of backward Euler needs to take into account the existence of potentially non-smooth projection operations, $\mathcal{Z}_E$ and $\mathcal{Z}_N$. The analytic form of the potential energy is, however, challenging to express (although possible with Ortiz and Stainier’s variational formulation [1999]). By noticing inelastic force balance comes from the stress evaluated on the projected strain, we can integrate the stress to define integral energy density functions $\Xi_E = \int \frac{\partial \Psi_E}{\partial \mathbf{F}_E}(\mathcal{Z}_E(\mathbf{F}_E^{\text{tr}})) d\mathbf{F}_E$ and $\Xi_N = \int \frac{\partial \Psi_N}{\partial \mathbf{F}_N}(\mathcal{Z}_N(\mathbf{F}_N^{\text{tr}})) d\mathbf{F}_N$ for both the equilibrium part and the non-equilibrated part, while their gradients would be the corresponding equilibrium force and non-equilibrated force. Then our problem can be described as

$$\begin{aligned} & \underset{\delta \mathbf{v}, \mathbf{F}_{\star}^{\text{tr}}}{\operatorname{argmin}} \quad \frac{1}{2} \delta \mathbf{v}^T \mathbf{M} \delta \mathbf{v} + \sum_p \sum_{\star \in \{E, N\}} V_p^0 \Xi_{\star, p} - \Delta t \delta \mathbf{v}^T \mathbf{M} \mathbf{g} \\ & \text{subject to} \quad \mathbf{W}_{\star}(\mathbf{F}_{\star}^{\text{tr}} - \mathbf{D} \delta \mathbf{v} - \mathbf{b}) = 0, \quad \star \in \{E, N\} \end{aligned} \quad (9)$$

where $\delta \mathbf{v}$ represents the MPM nodal velocity increment from t^n to t^{n+1} , $\mathbf{M}$ is the nodal mass matrix, $\mathbf{g}$ is gravitational acceleration, Δt is the time step, and $\mathbf{W}_\star$ are constraint weights. $\Xi_{\star,p}$ encodes particle p 's contribution to the objective function. The definitions of $\mathbf{D}$ and $\mathbf{b}$ become clear when we write down how the deformation gradient is evolved in standard MPM: $\mathbf{F}_\star^{\text{tr}} = (\mathbf{I} + dt \sum_i \delta \mathbf{v} (\nabla \mathbf{w}_{ip}^n)^T) \mathbf{F}_\star^n = \mathbf{D} \delta \mathbf{v} + \mathbf{b}$. $\mathbf{D}$ and $\mathbf{D}^T$ act as the bridges connecting node-centric quantities and particle-centric quantities. The two constraints are completely independent, even though both $\mathbf{F}_\star^{\text{tr}}$ are related to $\delta \mathbf{v}$ in the same way. The corresponding Lagrangian of problem (9) can be written as $L = \frac{1}{2} \delta \mathbf{v}^T \mathbf{M} \delta \mathbf{v} - \Delta t \delta \mathbf{v}^T \mathbf{M} \mathbf{g} + \sum_{\star \in \{E, N\}} \int \frac{\partial \Psi_\star}{\partial \mathbf{F}_\star} (\mathbf{Z}_\star(\mathbf{F}_\star^{\text{tr}})) d\mathbf{F}_\star + \sum_{\star \in \{E, N\}} \mathbf{y}_\star : [\mathbf{W}_\star(\mathbf{F}_\star - \mathbf{D} \delta \mathbf{v} - \mathbf{b})]$ and similarly, so can the augmented Lagrangian: $L_\rho = L + \frac{\rho}{2} \sum_{\star \in \{E, N\}} \|\mathbf{W}_\star(\mathbf{F}_\star - \mathbf{D} \delta \mathbf{v} - \mathbf{b})\|_F^2$. $\mathbf{y}_\star$ are the dual variables while ρ is the penalty coefficient.

We denote $\tilde{\mathbf{F}}$ as the concatenation of $\mathbf{F}_\star$, and $\tilde{\mathbf{y}}$ as the concatenation of $\mathbf{y}_\star$. ADMM alternates the updates of $\tilde{\mathbf{F}}$, $\delta \mathbf{v}$ and $\tilde{\mathbf{y}}$. We use notation $\tilde{\mathbf{F}}^{\text{tr}}$ instead of $\tilde{\mathbf{F}}^{n+1}$ to avoid possible confusions.

Particle $\mathbf{F}_{\star,p}$ step. We solve for both $\mathbf{F}_{\star,p}$ for each particle p by minimizing $L_\rho(\tilde{\mathbf{F}}, \delta \mathbf{v}^n, \tilde{\mathbf{y}}^n)$, i.e. solving $\frac{\partial L_\rho(\tilde{\mathbf{F}}, \delta \mathbf{v}^n, \tilde{\mathbf{y}}^n)}{\partial \mathbf{F}_{\star,p}} = 0$, to get

$$V_p^0 \frac{\partial \Psi_{\star,p}}{\partial \mathbf{F}_{\star,p}} (\mathbf{Z}_\star(\mathbf{F}_{\star,p}^{\text{tr}})) + \rho \mathbf{W}_{\star,p}^T \mathbf{W}_{\star,p} \mathbf{F}_{\star,p}^{\text{tr}} = \mathcal{R}_{\star,p} \quad (10)$$

where $\mathcal{R}_{\star,p} = \rho \mathbf{W}_{\star,p}^T \mathbf{W}_{\star,p} (\mathbf{D} \delta \mathbf{v}^n + \mathbf{b})_p - \mathbf{W}_{\star,p}^T \mathbf{y}_{\star,p}^n$.

To solve this problem, we take advantage of the property that the solution $\mathbf{F}_{\star,p}^{\text{tr},*}$ shares the same singular vectors with the right hand side [Klár et al. 2016; Overby et al. 2017] to further simplify the non-linear and non-smooth problem into just a d -dimensional problem in the diagonal space for each single particle. Together with the augmented term that essentially blends the potentially ill-conditioned force derivative term with a positive definite diagonal matrix at a similar scale, the system matrix also becomes much better conditioned, making it practical to solve via Newton's method in large time steps even without line search.

Specifically, for the k th Newton step, in the diagonal space, we can compute the Hessian as

$$V_p^0 \frac{\partial^2 \Psi_{\star,p}}{\partial \Sigma_{\star,p} \partial \tilde{\mathbf{Z}}_\star} (\tilde{\mathbf{Z}}_\star(\Sigma_{\star,p}^k)) \frac{\partial \tilde{\mathbf{Z}}_\star(\Sigma_{\star,p}^k)}{\partial \Sigma_{\star,p}} + \rho \mathbf{W}_{\star,p}^T \mathbf{W}_{\star,p}$$

which can be directly used to compute the particle Newton search direction. Note that due to the existence of $\frac{\partial \tilde{\mathbf{Z}}_\star(\Sigma_{\star,p}^k)}{\partial \Sigma_{\star,p}}$, the first term is asymmetric and may become singular, e.g. when cone tip projection happens in the Drucker-Prager plasticity model [Klár et al. 2016].

Similarly, in the Newton-based implicit plasticity solver by Klár et al. [2016], where the stiffness matrix is constructed on the background grid, the corresponding Hessian also includes two terms: the first term is closely related to ours and can be singular in certain projection cases. However, their second term is determined by grid-nodal masses and some of them can be quite small for the grid nodes that are in the vicinity of the boundary where particle coverage is sparse. On the other hand, our second term is entirely based on the uniform penalty parameter, ρ , and the independent

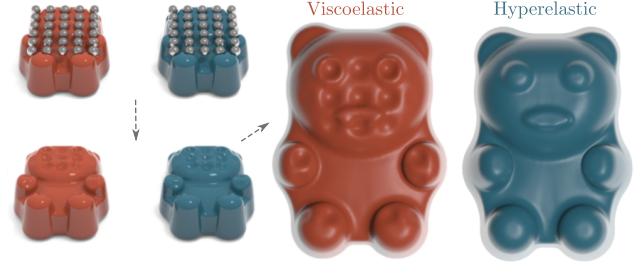


Fig. 9. **Panda.** An array of balls are used to compress two pandas simulated with viscoelasticity and hyperelasticity. While the hyperelastic panda immediately recovers its shape, the viscoelastic one takes much longer time to do so.

weighting function, $\mathbf{W}_{\star,p}$, offering better conditioning of the small system.

Grid $\delta \mathbf{v}$ step. The global step solves a large linear system for nodal velocity changes $\delta \mathbf{v}^{n+1}$:

$$\mathbf{A} \delta \mathbf{v}^{n+1} = \Delta t \mathbf{M} \mathbf{g} + \mathbf{D}^T \sum_{\star \in \{E, N\}} \left(\mathbf{W}_\star^T \mathbf{y}_\star^n + \rho \mathbf{W}_\star^T \mathbf{W}_\star (\mathbf{F}_\star^{\text{tr}} - \mathbf{b}) \right),$$

where $\mathbf{A} = \mathbf{M} + \rho \mathbf{D}^T (\sum_{\star \in \{E, N\}} \mathbf{W}_\star^T \mathbf{W}_\star) \mathbf{D}$ is SPD and can be solved by Conjugate Gradient. We found that a diagonal preconditioner performs sufficiently well. In practice we choose to store the matrix since it is fixed throughout the time step. We observe that the cost of building and using the matrix is significantly lower than performing costly multiplication as in matrix-free Krylov solvers.

Dual variable update. We add a standard quadratic regularizer to the augmented Lagrangian to solve for the dual steps as

$$\mathbf{y}_{\star,p}^{n+1} = \mathbf{y}_{\star,p}^n + \rho \mathbf{W}_{\star,p} (\mathbf{F}_{\star,p}^{\text{tr}} - \mathbf{D} \delta \mathbf{v}^{n+1} - \mathbf{b})$$

following Overby et al. [2017].

7.2 Boundary Conditions

We only need to enforce the boundary condition for the grid nodal velocities (without worrying about the deformation gradient and the dual variable). At the beginning of a time step, we first detect and record the grid nodes that are in contact with collision objects, and then ensure the initial guess of $\delta \mathbf{v}^{n+1}$ to satisfy the boundary condition. During each particle step, we project inside the CG solver to prevent any further changes in $\delta \mathbf{v}^{n+1}$.

7.3 Improving Convergence

Stiffness Enhanced Weighting (SEW). It is well known that constraint weighting is essential to ADMM convergence [Boyd et al. 2011]. Inappropriate weighting strategies will result in slow or, potentially, no convergence at all. Overby et al. [2017] weight the constraint of each mesh element using a constant bulk modulus throughout the full simulation. While working well in most of their cases, this strategy does not account for the current deformation state and may perform less effectively during large deformations. We propose a Stiffness Enhanced Weighting (SEW) that sets $\mathbf{W}_\star$

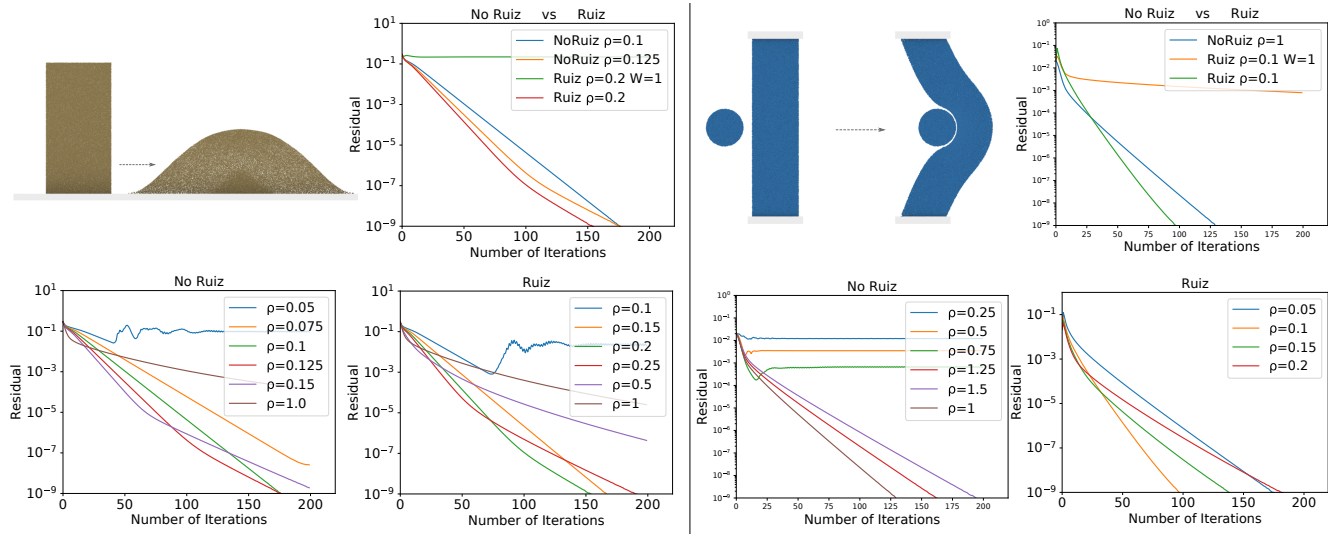


Fig. 10. **Ruiz Equilibration.** For the elastic case on the right, No-Ruiz needs fewer iterations than Ruiz to converge to a residual of $1e^{-4}$ due to the effectiveness of our SEW strategy (although they would intersect as more and more iterations were run). On the other hand, for the inelastic sand model on the left, Ruiz outperforms No-Ruiz from the beginning since SEW is known not to work well for inelastic materials due to the non-smoothness of the flow rule. We also show that ρ , interpreted as the uniform scaling parameter, plays a key role in the overall convergence of ADMM. We have handpicked the best corresponding ρ values for both Ruiz and No-Ruiz results for a fair comparison. Sophisticated versions of residual balancing [Wohlberg 2017] for automatically tuning ρ will be investigated as future work.

as $\sqrt{\|\frac{\partial^2 \Psi_\star(\Sigma_\star)}{\partial \Sigma_\star^2}(\Sigma_\star)\|_F}$ for pure elasticity at the beginning of each time step to provide more accurate second-order information to our ADMM solver to improve the convergence as shown in Fig. 11. For inelastic models, $\frac{\partial \tilde{Z}_\star(\Sigma_\star)}{\partial \Sigma_\star}$ also needs to be taken into consideration; however, when this derivative is zero, e.g., projecting to the tips in Drucker-Prager, we use the unprojected version to avoid the possible singularity.

Ruiz Equilibration and Residual Balancing. We further enhance the convergence by exploring both Ruiz equilibration [Stellato et al. 2018] and residual balancing [Wohlberg 2017]. Ruiz equilibration is derived by first constructing the KKT system with both primal and dual variables considered, and then computing the scaling variables

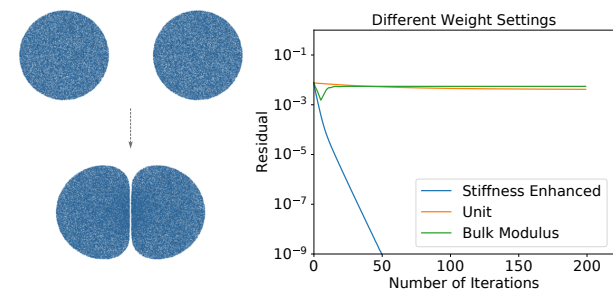


Fig. 11. **Elastic Balls.** Our stiffness enhanced weight significantly expedites the convergence while both the bulk modulus weight [Overby et al. 2017] and the unit weight fail to achieve similar performance.

that can equilibrate the norm of each row/column of the KKT system to achieve a better condition number. Note that Stellato et al. [2018] applied Ruiz equilibration to a simple quadratic problem; in our case, we first linearize the portions related to the force derivatives at the previous time step. For efficiency considerations, we compute Ruiz equilibration once per time step. We include the derivation and pseudo-code for Ruiz equilibration in the supplemental document [Fang et al. 2019].

We present a convergence study for Ruiz equilibration in Fig. 10 for representative time instances with large deformation in each simulated scene. For pure elasticity, our Ruiz equilibration fails to outperform the non-Ruiz version augmented with our SEW, demonstrating SEW’s efficacy. However, as mentioned, for inelasticity, with carefully tuned ρ , Ruiz manages to deliver better improvements than

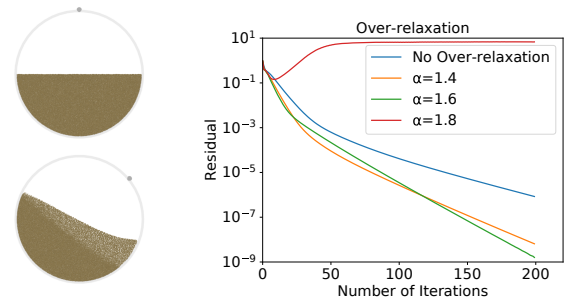


Fig. 12. **Roller.** While two cases of over-relaxations (with $\alpha = 1.4$ and $\alpha = 1.6$) achieve performance benefits over the one without over-relaxation, an even larger α turns out to be detrimental.

the non-Ruiz version since non-smoothness can easily ruin the effectiveness of SEW since occasionally unprotected stiffness is used.

Residual balancing is a heuristic method and helps to improve the convergence in many problems by reinterpreting the penalty coefficient ρ as a critical tool to balance the primal residual and dual residual, such that neither of them would end up expanding arbitrarily. In our experiments, a relatively simple version does not seem to improve the convergence. We leave a more promising alternative [Wohlberg 2017] as future work. In practice, manual tuning of ρ can still produce significantly better convergence behavior as shown in Fig. 10.

Over-Relaxation. Another set of strategies for improving the convergence of first-order methods is to utilize the information of previous iterates to construct a better approximation to the current local energy manifold. Methods like Anderson Acceleration [Peng et al. 2018; Walker and Ni 2011] and Nesterov Acceleration [Nesterov et al. 2007] have all been shown to work effectively on unconstrained optimization scenarios.

For the constrained solvers that explicitly handle dual variables, e.g. ADMM, Boyd et al. [2011] suggest a simple scheme named over-relaxation [Eckstein 1994] which looks back by one previous iterate and applies it to the global and dual steps. In the grid $\delta\mathbf{v}$ step, we change $\mathbf{F}_\star^{\text{tr}}$ in the right hand side to $\alpha\mathbf{F}_\star^{\text{tr}} + (1-\alpha)(D\delta\mathbf{v}^n + \mathbf{b})$. In the ADMM literature [Boyd et al. 2011], $\alpha \in [1, 2]$ usually accelerates convergence, and $\alpha \in [1.5, 1.8]$ has been shown to work remarkably well on specific convex problems. Since our problem is non-convex and non-smooth, the working range of the parameter can be slightly different and requires some tolerable manual exploration. Furthermore, in the dual step, although now $\delta\mathbf{v}^{n+1}$ is available, the same replacement should be applied to ensure consistency.

8 TRANSFER-BASED DAMPING

Here we describe an extremely simple yet useful method for introducing additional artificial damping into MPM simulations. The technique described in this section is independent of the type of material being simulated and works for any time integration schemes.

Our scheme is based on the Rigid Particle-In-Cell (RPIC) method [Jiang et al. 2015]. As pointed out by Jiang et al. [2017], RPIC can be thought of as a reduced APIC scheme that suppresses stretching and shearing velocity modes. Here we assume the usage of MLS-MPM force computation [Hu et al. 2018] and rigorously demonstrate the equivalence of RPIC and physical viscosity. If Affine Particle-In-Cell (APIC) is chosen as the default transfer scheme, RPIC can be constructed by only keeping the skew-symmetric (rotational) part of the velocity gradient matrix $\mathbf{C}_p^n$ before performing particle-to-grid transfers. This can be written as $(m\mathbf{v})_i^n = \sum_p m_p w_{ip}^n \left(\mathbf{v}_p^n + (\mathbf{C}_p^n - \mathbf{C}_p^{nT})/2(\mathbf{x}_i - \mathbf{x}_p^n) \right)$, which can be additively decomposed into $\sum_p m_p w_{ip}^n \left(\mathbf{v}_p^n + \mathbf{C}_p^n(\mathbf{x}_i - \mathbf{x}_p^n) \right)$ (APIC contribution) and an additional impulse $\mathcal{I}_i$ where

$$\mathcal{I}_i = - \sum_p m_p w_{ip}^n (\mathbf{C}_p^n + \mathbf{C}_p^{nT})/2(\mathbf{x}_i - \mathbf{x}_p^n). \quad (11)$$

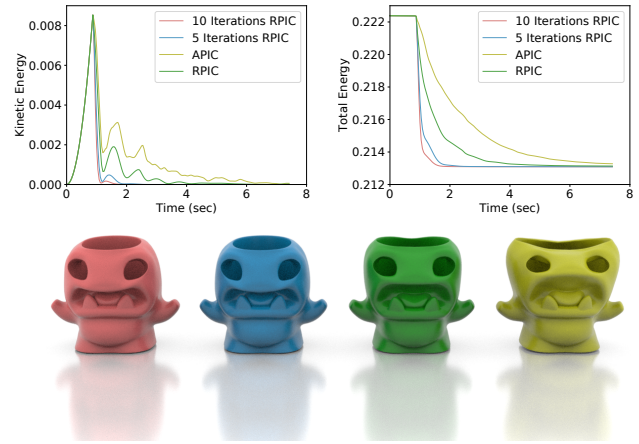


Fig. 13. **Alien.** Our iterative damping strategy easily introduces damping in a controllable way by setting the number of RPIC iterations.

Comparing this with the MLS-MPM explicit force [Hu et al. 2018] $\mathbf{f}_i = -\sum_p V_p^n \frac{4}{\Delta x^2} \sigma_p^n w_{ip}^n (\mathbf{x}_i - \mathbf{x}_p^n)$, we can observe that the impulse is equivalent to applying a force resulting from a Cauchy stress $\sigma_p^n = \frac{\rho_p^n \Delta x^2}{4\Delta t} \frac{\mathbf{C}_p^n + \mathbf{C}_p^{nT}}{2}$. By noticing that $\mathbf{C}_p^n$ is a discrete velocity gradient evaluation, such a stress corresponds to a standard Newtonian viscosity with kinematic viscosity of value $\nu = \frac{1}{8} \frac{\Delta x^2}{\Delta t}$, which turns out to exactly match the diffusive stability constraint $\nu\Delta t/\Delta x^2 \leq (1/2)^3$ [Johnston and Liu 2004] for high Reynolds number viscous fluids at the critical point.

In other words, we demonstrated that going from APIC to RPIC is equivalent to performing Newtonian viscosity *explicitly* without breaking the requirement for numerical stability. Given this appealing theoretical ground, we could introduce an arbitrary amount of unconditionally stable numerical damping by performing multiple RPIC velocity transfer round-trips between particles and the grid (without particle advection or strain update) as a preprocessing routine in each time step. Due to the conservative properties of RPIC, such a numerical damping scheme is not only easy to implement and can be used in any MPM integrator, but also precisely conserves linear and angular momentums. In Fig. 13, we compare the effect of RPIC damping iterations on the simulation of a dropping elastic object. We believe continuous control of damping is also possible by artificially adjusting the Δt value in the virtual RPIC round-trips. More in-depth investigation of fully controllable damping is left for future work.

It is noteworthy that Newtonian viscosity can also be directly incorporated into the conservation of momentum as done by Ram et al. [2015]. Within our ADMM framework, it corresponds to an additional term in the global system. We found the transfer-based damping more useful in practice due to its guaranteed stability (especially for explicit integration), ease of implementation, and computational efficiency.

Table 2. **Simulation parameters and statistics.** Particle count and time per frame are provided as average values. All numbers are measured on an Intel Core i9-9900K CPU with 8 cores at 3.60 GHZ.

Example	Sec/Frame	Δx	Δt	N	ρ	μ_E	λ_E	μ_N	λ_N	ν
(Fig. 1) Cherries	228.5	5.0×10^{-3}	2.0×10^{-3}	0.43M	2	$3.4 \times 10^{-3} / 1.4 \times 10^{-2}$	2.9×10^1	2.1×10^0	8.6×10^0	$4.8 \times 10^{-3} / 2.0 \times 10^{-2}$
(Fig. 3) Warrior	122.6	6.0×10^{-3}	4.0×10^{-3}	0.52M	2	3.6×10^{-3}	1.4×10^2	7.1×10^0	2.9×10^1	2.0×10^{-1}
(Fig. 7) Twisting Metal	58.4	1.0×10^{-2}	5.0×10^{-3}	0.20M	2	7.1×10^2	2.9×10^3	-	-	-
(Fig. 7) Twisting Elastic	29.4	1.0×10^{-2}	1.0×10^{-2}	0.20M	2	3.6×10^1	1.4×10^2	-	-	-
(Fig. 5) Silicone Rubber	82.1	7.5×10^{-3}	8.5×10^{-3}	0.65M	2	1.8×10^0	7.1×10^1	1.8×10^1	7.1×10^1	1.0×10^1
(Fig. 9) Panda	51.1	1.0×10^{-2}	8.5×10^{-5}	0.42M	2	1.8×10^0	7.1×10^1	1.8×10^1	7.1×10^1	3.0×10^0
(Fig. 14) Coil	26.0	1.0×10^{-2}	4.0×10^{-3}	0.16M	2	3.5×10^{-3}	1.4×10^3	7.1×10^0	2.9×10^1	4.0×10^{-1}
(Fig. 15) Pet Monster	21.9	1.0×10^{-2}	1.1×10^{-2}	0.20M	2	3.8×10^{-1}	5.8×10^0	3.8×10^0	5.8×10^0	5.0×10^{-2}
(Fig. 1) Faceless	41.1	1.5×10^{-2}	1.4×10^{-2}	0.42M	2	1.8×10^0	7.1×10^1	1.8×10^1	7.1×10^1	1.0×10^1
(Fig. 2) Car	64.2	2.0×10^{-2}	3.5×10^{-3}	0.22M	2	7.1×10^2	2.9×10^3	-	-	-
(Fig. 17) Sofa	574.1	3.7×10^{-3}	3.5×10^{-3}	1.57M	2	3.6×10^{-1}	1.4×10^2	3.6×10^1	1.4×10^2	3.0×10^{-1}

9 RESULTS

We implement our method on a high-performance CPU-based MPM code base (to be open-sourced on the authors' homepages). We also attach the pseudo-codes for our implicit inelasticity solver in the supplemental document. Note that nothing is preventing a straightforward integration of our method into a GPU MPM pipeline, e.g. that of Gao et al. [2018b]. All of our simulation statistics are measured on an Intel Core i9-9900K CPU. For rendering fluid-like materials, we reconstruct the surface from the particles using OpenVDB [Museth et al. 2013]. For solid-like materials without topology change, we embed a high resolution surface [Wojtan et al. 2009] in the original particle samples, and deform it with the deformation flow. The embedded surface gracefully retains the sharp and flat features without requiring any post-processing such as Gaussian smoothing. In practice, we solve global steps with high tolerances and use a fixed number of ADMM iterations for efficiency considerations. We adopt the Stiffness Enhanced Weighting in all examples. Ruiz and over-relaxation require additional computational cost that does not compensate for the convergence improvement they bring in high-resolution 3D cases. Higher performance utilization of these is left to future work.

Controlling the look. By varying the equilibrated shear modulus μ_E (Fig. 14, Fig. 1), we can control the fluidity of the simulated matter. In particular, $\mu_E = 0$ will degrade the equilibrated hyperelasticity model into a nearly incompressible fluid. As such, smaller μ_E results in a more fluid-like look, and larger μ_E corresponds to a more elastic (and shape-preserving) behavior. Note that unlike plasticity

Equilibrated shear modulus (μ_E)

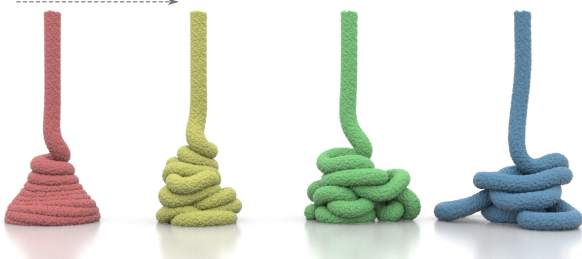


Fig. 14. **Coil.** Our model can easily capture coil buckling behavior as shown on the left. From left to right the equilibrated shear modulus coefficient increases and coils behave more and more elastically.

or viscoplasticity which tend to trigger permanent changes of the rest shape, viscoelastic materials always retain an invariant rest shape. However, when μ_E is too small to resist large distortions or topological changes, MPM automatically complies with a geometrically plausible shape without trying to pull the material back to its original shape. This can be illustrated by dropping a long and thick viscoelastic sheet onto a rotating statue, producing complex folding behaviors (Fig. 3). Similarly, viscosity parameters ν_v and ν_d control the intensity of the viscous force. We pick the same value for simplicity. Objects with larger viscosity have greater resistance to rapid motion as shown in Fig. 15.

Traditional elastoplasticity can also be easily integrated into our framework (Fig. 7, 2). Our ADMM solver generates similar results to those of explicit MPM solvers. Our framework can also be applied to Drucker-Prager materials as demonstrated in Fig. 16. For handling high stiffness models (e.g. sand) explicit time integration has to work with a small time step. On the other hand, traditional implicit solvers, e.g., the Newton solver used in [Klár et al. 2016], rely on GMRES for solving a large asymmetric system, proving to be even slower than explicit alternatives even with a relatively large time step. Our framework can generate visually plausible results without requiring stringently accurate convergence. As a result, our solver stays stable even when the time step is larger than what is permitted in [Klár et al. 2016] and achieves high performance.

Sand has also been popularly modeled with other continuum approaches, for instance, as a unilaterally incompressible viscoplastic medium [Narain et al. 2010], in which case the adaptation of our framework (for incompressibility) demands future exploration.

Viscosity (ν_d and ν_v)

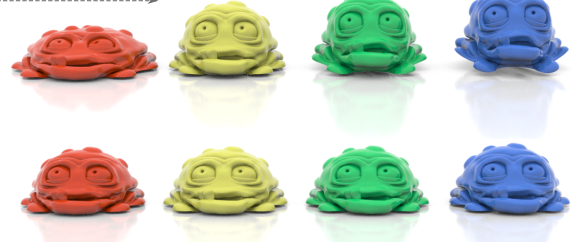


Fig. 15. **Petmon.** From left to right, we demonstrate that larger ν_d and ν_v show more resistance to high rate deformation.

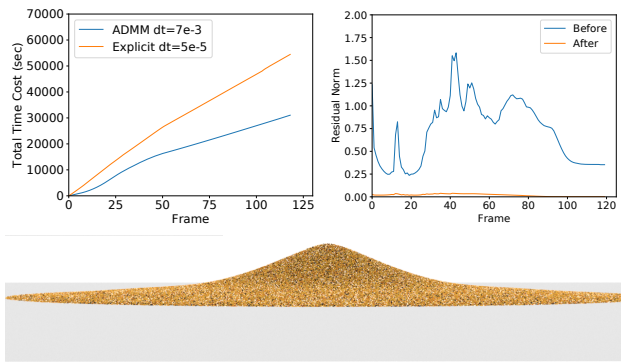


Fig. 16. **Sand column collapse.** Our ADMM solver not only captures characteristic sand column collapse dynamics, but also improves the overall performance since our solver permits a much larger time step. Left shows timing comparison against explicit integration, and right shows nonlinear residual norm decrease (averaged per frame) with 15 ADMM iterations.

The work by Daviet and Bertails-Descoubes [2016] also treats sand as a nonsmooth viscoplastic granular media and captures implicit friction by leveraging nonsmooth optimization techniques in MPM. We performed the sand column collapse test (Fig. 16) with Daviet’s open-source code, and observed 40% better performance than ours.

Viscoelastic solids can slowly recover to the original shape after severe deformations (Fig. 5). We make comparisons between our viscoelasticity model and traditional hyperelasticity model as on the left of Fig. 1 and as in Fig. 9. While the hyperelastic objects immediately recover the default shape, the viscoelastic ones take much longer to do so. Our framework can also automatically produce detailed fractures when a collision cylinder moves and collides with a solid foam-like sofa as in Fig. 17.

10 LIMITATIONS AND FUTURE WORK

In this paper, we presented a framework for simulating viscoelastic solids and fluids with MPM. Our formulation results in a strain projection algorithm that is remarkably similar to the standard return mapping for plasticity, and can be easily integrated into any existing MPM framework. To enable fully implicit integration with backward Euler we extend the recently advanced ADMM from FEM hyperelasticity to MPM inelasticity. Inelasticity prohibits the existence of an easily evaluable objective function. Nonetheless, we show that the variable splitting approach with the updated Lagrangian results in a robust and efficient solver.

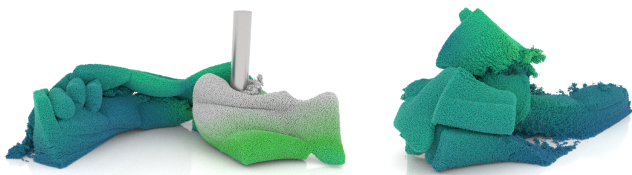


Fig. 17. **Sofa.** Two solid foam sofas get torn apart and come to rest, showing the intricate topology change automatically handled by MPM.

While allowing visually plausible results, our formulation inherits the limitations of ADMM itself. ADMM belongs to the family of first-order methods. Even though it can rapidly reduce the residual in the first few iterations, it soon gets stuck in a slow convergence zone. Furthermore, most applications that favor ADMM have no stringent demands for high accuracy. However, for certain hard constraints such as incompressibility and inextensibility, low accuracy would render unacceptable consequences. This restriction is challenging but worth solving, and we would like to investigate possible solutions (such as Nesterov-type acceleration [Goldstein et al. 2014]) as our future work.

Our framework also inherits limitations from MPM. For example, even though elastic volume change is captured by the elastic deformation (and the inelasticity projection is volume preserving), perfect visual volume preservation is not automatically guaranteed due to the numerical errors in advection. We would like to explore improving particle distribution as future work.

In contrast to the return mapping-based approaches, hyperplasticity [Houlsby and Puzrin 2007] provides a thermodynamically consistent reformulation of elastoplasticity as a clearly defined optimization problem following the second law of thermodynamics. We would like to explore efficient options under these principles. Extending our current formulation to viscoplasticity to capture effects such as in Yue et al. [2015] is also an exciting direction. Finally, we would like to include quantitative validations with real data to study the predictivity of our framework in computational science.

ACKNOWLEDGMENTS

We would like to thank Tiantian Liu and Danny Kaufman for useful discussions, Joshua Wolper for proofreading and narrating the video, and the anonymous reviewers for their valuable comments. This work was supported in part by NSF Grants IIS-1755544 and CCF-1813624, a gift from Adobe Inc., NVIDIA GPU grants, and Houdini licenses from SideFX.

REFERENCES

- A. Bargteil, C. Wojtan, J. Hodgins, and G. Turk. 2007. A finite element method for animating large viscoplastic flow. *ACM Trans Graph* 26, 3 (2007).
- H. Barreiro, I. García-Fernández, I. Alduán, and M. Otaduy. 2017. Conformation constraints for efficient viscoelastic fluid simulation. *ACM Trans Graph* 36, 6 (2017).
- C. Batty, F. Bertails, and R. Bridson. 2007. A fast variational framework for accurate solid-fluid coupling. *ACM Trans Graph* 26, 3 (2007).
- C. Batty and R. Bridson. 2008. Accurate viscous free surfaces for buckling, coiling, and rotating liquids. *Proc ACM SIGGRAPH/Eurograph Symp Comp Anim* (2008), 219–228.
- C. Batty and B. Houston. 2011. A simple finite volume method for adaptive viscous liquids. In *Symp on Comp Anim*. 111–118.
- C. Batty, A. Uribe, B. Audoly, and E. Grinspun. 2012. Discrete viscous sheets. *ACM Trans Graph* 31, 4 (2012), 113.
- J. Bonet and R. Wood. 2008. *Nonlinear continuum mechanics for finite element analysis*. S. Bouaziz, S. Martin, T. Liu, L. Kavan, and M. Pauly. 2014. Projective dynamics: fusing constraint projections for fast simulation. *ACM Trans Graph* 33, 4 (2014), 154.
- S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning* 3, 1 (2011), 1–122.
- G. Brown, M. Overby, Z. Forootaninia, and R. Narain. 2018. Accurate dissipative forces in optimization integrators. In *SIGGRAPH Asia 2018 Papers*.
- M. Carlson, P. Mucha, R.B. Van Horn, and G. Turk. 2002. Melting and Flowing. In *Proc of the 2002 ACM SIGGRAPH/EuroGraph Symp on Comp Anim (SCA '02)*. 167–174.
- N. Chentanez, M. Müller, and M. Macklin. 2016. Real-time simulation of large elastoplastic deformation with shape matching. In *Symp on Comp Anim*. 159–167.
- G. Daviet and F. Bertails-Descoubes. 2016. A semi-implicit material point method for the continuum simulation of granular materials. *ACM Trans Graph* 35, 4 (2016), 102:1–102:13.

- D. Dinev, T. Liu, and L. Kavan. 2018a. Stabilizing Integrators for Real-Time Physics. *ACM Trans. Graph.* (2018).
- D. Dinev, T. Liu, J. Li, B. Thomaszewski, and L. Kavan. 2018b. FEPR: Fast Energy Projection for Real-time Simulation of Deformable Objects. *ACM Trans. Graph.* 37, 4 (July 2018), 79:1–79:12.
- J. Eckstein. 1994. Parallel alternating direction multiplier decomposition of convex programs. *J of Optimization Theory and Applications* 80, 1 (1994), 39–62.
- Y. Fang, Y. Hu, S. Hu, and C. Jiang. 2018. A temporally adaptive material point method with regional time stepping. In *Comp Graph forum*, Vol. 37. 195–204.
- Y. Fang, M. Li, M. Gao, and C. Jiang. 2019. Silly Rubber: supplemental document. (2019).
- Y. Fei, C. Batty, E. Grinspun, and C. Zheng. 2018. A Multi-scale Model for Simulating Liquid-fabric Interactions. *ACM Trans. Graph.* 37, 4 (Aug. 2018), 51:1–51:16.
- M. Fortin and A. Fortin. 1989. A new approach for the FEM simulation of viscoelastic flows. *J of non-newtonian fluid Mech* 32, 3 (1989), 295–310.
- M. Fortin and R. Glowinski. 1983. Chapter III on decomposition-coordination methods using an augmented lagrangian. In *Studies in Math and Its Appl.* Vol. 15. 97–146.
- M. Gao, A. Pradhana, X. Han, Q. Guo, G. Kot, E. Sifakis, and C. Jiang. 2018a. Animating fluid sediment mixture in particle-laden flows. *ACM Trans Graph* 37, 4 (2018), 149.
- M. Gao, A. Pradhana, C. Jiang, and E. Sifakis. 2017. An adaptive generalized interpolation material point method for simulating elastoplastic materials. *ACM Trans Graph* 36, 6 (2017), 223.
- M. Gao, X. Wang, K. Wu, A. Pradhana, E. Sifakis, C. Yuksel, and C. Jiang. 2018b. GPU Optimization of Material Point Methods. *ACM Trans Graph* (2018), 254:1–254:12.
- T. Gast, C. Schroeder, A. Stomakhin, C. Jiang, and J. Teran. 2015. Optimization Integrator for Large Time Steps. *IEEE Trans Vis Comp Graph* 21, 10 (2015), 1103–1115.
- J. Gaume, T. Gast, J. Teran, A. van Herwijnen, and C. Jiang. 2018. Dynamic anticrack propagation in snow. *Nature Comm* 9, 1 (2018), 3047.
- D. Gerszewski, H. Bhattacharya, and A. Bargteil. 2009. A point-based method for animating elastoplastic solids. In *Symp on Comp Anim.* 133–138.
- T. Goktekin, A. Bargteil, and J. O'Brien. 2004. A method for animating viscoelastic fluids. *ACM Trans Graph* 23, 3 (2004), 463–468.
- T. Goldstein, B. O'Donoghue, S. Setzer, and R. Baraniuk. 2014. Fast alternating direction optimization methods. *SIAM J on Imag Sci* 7, 3 (2014), 1588–1623.
- S. Govindjee and S. Reese. 1997. A presentation and comparison of two large deformation viscoelasticity models. *J of Eng Mat and technology* 119, 3 (1997), 251–255.
- Q. Guo, X. Han, C. Fu, T. Gast, R. Tamstorf, and J. Teran. 2018. A material point method for thin shells with frictional contact. *ACM Trans Graph* 37, 4 (2018), 147.
- X. He, H. Wang, and E. Wu. 2018. Projective peridynamics for modeling versatile elastoplastic materials. *IEEE Trans Vis and Comp Graph* 24, 9 (2018), 2589–2599.
- G.T. Houlsby and A.M. Puzrin. 2007. *Principles of hyperplasticity: an approach to plasticity theory based on thermodynamic principles.*
- Y. Hu, Y. Fang, Z. Ge, Z. Qu, Y. Zhu, A. Pradhana, and C. Jiang. 2018. A moving least squares material point method with displacement discontinuity and two-way rigid body coupling. *ACM Trans Graph* 37, 4 (2018), 150.
- G. Irving, J. Teran, and R. Fedkiw. 2004. Invertible finite elements for robust simulation of large deformation. In *Proc ACM SIGGRAPH/Eurograph Symp Comp Anim.* 131–140.
- C. Jiang, T. Gast, and J. Teran. 2017. Anisotropic elastoplasticity for cloth, knit and hair frictional contact. *ACM Trans Graph* 36, 4 (2017).
- C. Jiang, C. Schroeder, A. Selle, J. Teran, and A. Stomakhin. 2015. The affine particle-in-cell method. *ACM Trans Graph* 34, 4 (2015), 51:1–51:10.
- H. Johnston and J. Liu. 2004. Accurate, stable and efficient Navier–Stokes solvers based on explicit treatment of the pressure term. *J of Comp Phys* 199, 1 (2004), 221–259.
- B. Jones, S. Ward, A. Jallepalli, J. Perenia, and A. Bargteil. 2014. Deformation embedding for point-based elastoplastic simulation. *ACM Trans Graph* 33, 2 (2014), 21.
- G. Klár, T. Gast, A. Pradhana, C. Fu, C. Schroeder, C. Jiang, and J. Teran. 2016. Drucker-prager elastoplasticity for sand animation. *ACM Trans Graph* 35, 4 (2016).
- E. Larionov, C. Batty, and R. Bridson. 2017. Variational stokes: a unified pressure-viscosity solver for accurate viscous liquids. *ACM Trans Graph* 36, 4 (2017), 101.
- Patrick Le Tallec. 1990. *Numerical analysis of viscoelastic problems.* Vol. 15.
- J. Li, T. Liu, and L. Kavan. 2018. Laplacian Damping for Projective Dynamics. In *VRIPHYS2018: 14th Workshop on Virtual Reality Interaction and Physical Sim.*
- T. Liu, A. Bargteil, J. O'Brien, and L. Kavan. 2013. Fast Simulation of Mass-Spring Systems. *ACM Trans Graph* 32, 6 (2013), 209:1–7.
- T. Liu, S. Bouaziz, and L. Kavan. 2017. Quasi-newton methods for real-time simulation of hyperelastic materials. *ACM Trans Graph* 36, 4 (2017), 116a.
- F. Losasso, T. Shinar, A. Selle, and R. Fedkiw. 2006. Multiple Interacting Liquids. In *ACM SIGGRAPH 2006 Papers (SIGGRAPH '06)*. 812–819.
- A. Mielke. 2006. A mathematical framework for generalized standard materials in the rate-independent case. In *Multifield Problems in Solid and Fluid Mech.* 399–428.
- K. Museth, J. Lait, J. Johanson, J. Budsberg, R. Henderson, M. Alden, P. Cucka, D. Hill, and A. Pearce. 2013. OpenVDB: an open-source data structure and toolkit for high-resolution volumes. In *siggraph 2013 courses.*
- R. Narain, A. Golas, and M. Lin. 2010. Free-flowing granular materials with two-way solid coupling. *ACM Trans Graph* 29, 6 (2010), 173:1–173:10.
- B. Nedjar. 2002a. Frameworks for finite strain viscoelastic-plasticity based on multiplicative decompositions. Part I: Continuum formulations. *Comp Meth in App Mech and Eng* 191, 15–16 (2002), 1541–1562.
- B. Nedjar. 2002b. Frameworks for finite strain viscoelastic-plasticity based on multiplicative decompositions. Part II: Computational aspects. *Comp Meth in App Mech and Eng* 191, 15–16 (2002), 1563–1593.
- Y. Nesterov et al. 2007. Gradient methods for minimizing composite objective function.
- Michael Ortiz and Laurent Stainier. 1999. The variational formulation of viscoplastic constitutive updates. *Comp Meth in App Mech and Eng* 171, 3–4 (1999), 419–444.
- M. Overby, G. Brown, J. Li, and R. Narain. 2017. ADMM $\supseteq$ Projective Dynamics: Fast Simulation of Hyperelastic Models with Dynamic Constraints. *IEEE Trans. Vis. Comput. Graph.* 23, 10 (Oct. 2017), 2222–2234.
- Y. Peng, B. Deng, J. Zhang, F. Geng, W. Qin, and L. Liu. 2018. Anderson Acceleration for Geometry Optimization and Physics Simulation. *arXiv* (2018).
- A. Pradhana, T. Gast, G. Klár, C. Fu, J. Teran, C. Jiang, and K. Museth. 2017. Multi-species simulation of porous sand and water mixtures. *ACM Trans Graph* 36, 4 (2017).
- D. Ram, T. Gast, C. Jiang, C. Schroeder, A. Stomakhin, J. Teran, and P. Kavehpour. 2015. A material point method for viscoelastic fluids, foams and sponges. In *Proc ACM SIGGRAPH/Eurograph Symp Comp Anim.* 157–163.
- N. Rasmussen, D. Enright, D. Nguyen, S. Marino, N. Sumner, W. Geiger, S. Hoon, and R. Fedkiw. 2004. Directable photorealistic liquids. In *Proc of the 2004 ACM SIGGRAPH/EuroGraph Symp on Comp Anim.* 193–202.
- S. Reese and S. Govindjee. 1998. A theory of finite viscoelasticity and numerical aspects. *Int J of solids and structures* 35, 26–27 (1998), 3455–3482.
- E. Sifakis and J. Barbič. 2015. Finite Element Method Simulation of 3D Deformable Solids. *SIGGRAPH Course* 1, 1 (2015), 1–69.
- J. C. Simo. 1992. Algorithms for static and dynamic multiplicative plasticity that preserve the classical return mapping schemes of the infinitesimal theory. *Comp Meth App Mech Eng* 99, 1 (1992), 61–112.
- Juan C Simo and Thomas JR Hughes. 2006. *Computational inelasticity.* Vol. 7.
- B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd. 2018. OSQP: An Operator Splitting Solver for Quadratic Programs. In *Int Conf on Control.* 339–339.
- A. Stomakhin, C. Schroeder, L. Chai, J. Teran, and A. Selle. 2013. A material point method for snow simulation. *ACM Trans Graph* 32, 4 (2013), 102:1–102:10.
- A. Stomakhin, C. Schroeder, C. Jiang, L. Chai, J. Teran, and A. Selle. 2014. Augmented MPM for phase-change and varied materials. *ACM Trans Graph* 33, 4 (2014), 138.
- D. Sulsky, S. Zhou, and H. Schreyer. 1995. Application of a particle-in-cell method to solid mechanics. *Comp Phys Comm* 87, 1 (1995), 236–252.
- T. Takahashi, Y. Dobashi, I. Fujishiro, T. Nishita, and M. Lin. 2015. Implicit formulation for SPH-based viscous fluids. In *Comp Graph Forum*, Vol. 34. 493–502.
- T. Takahashi, T. Nishita, and I. Fujishiro. 2014. Fast simulation of viscous fluids with elasticity and thermal conductivity using position-based dynamics. *Comps Graph* 43 (2014), 21–30.
- J. Teran, E. Sifakis, G. Irving, and R. Fedkiw. 2005. Robust quasistatic finite elements and flesh simulation. In *Symp on Comp Anim.* 181–190.
- D. Terzopoulos and K. Fleischer. 1988. Modeling inelastic deformation: viscoelasticity, plasticity, fracture. In *ACM Siggraph Comp Graph*, Vol. 22. 269–278.
- D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. 1987. Elastically deformable models. *ACM Siggraph Comp Graph* 21, 4 (1987), 205–214.
- H.F. Walker and P. Ni. 2011. Anderson acceleration for fixed-point iterations. *SIAM J on Numer Analysis* 49, 4 (2011), 1715–1735.
- H. Wang and Y. Yang. 2016. Descent methods for elastic body simulation on the GPU. *ACM Trans Graph* 35, 6 (2016), 212.
- M. Wicke, D. Ritchie, B. Klingner, S. Burke, J. Shewchuk, and J. O'Brien. 2010. Dynamic local remeshing for elastoplastic simulation. *ACM Trans Graph* 29, 4 (2010), 49:1–11.
- B. Wohlberg. 2017. ADMM Penalty Parameter Selection by Residual Balancing. (2017).
- Chris Wojtan, Nils Thürey, Markus Gross, and Greg Turk. 2009. Deforming meshes that split and merge. In *ACM Trans Graph*, Vol. 28. 76.
- C. Wojtan and G. Turk. 2008. Fast viscoelastic behavior with thin features. *ACM Trans Graph* 27, 3 (2008), 47.
- J. Wrethorn, R. Armiento, and K. Museth. 2017. Animation of Crack Propagation by Means of an Extended Multi-body Solver for the Material Point Method. *Comput. Graph.* 69, C (Dec. 2017), 131–139. <https://doi.org/10.1016/j.cag.2017.10.005>
- X. Yan, C. Li, X. Chen, and S. Hu. 2018. MPM simulation of interacting fluids and solids. *Comp Graph Forum* 37, 8 (2018), 183–193.
- Y. Yue, B. Smith, C. Batty, C. Zheng, and E. Grinspun. 2015. Continuum foam: a material point method for shear-dependent flows. *ACM Trans Graph* 34, 5 (2015), 160:1–160:20.
- Y. Yue, B. Smith, P. Chen, M. Chantharayukhonthorn, K. Kamrin, and E. Grinspun. 2018. Hybrid Grains: Adaptive Coupling of Discrete and Continuum Simulations of Granular Media. *ACM Trans Graph* (2018), 283:1–283:19.
- B. Zhu, M. Lee, E. Quigley, and R. Fedkiw. 2015. Codimensional non-Newtonian fluids. *ACM Trans Graph* 34, 4 (2015), 115.
- Y. Zhu and R. Bridson. 2005. Animating sand as a fluid. *ACM Trans Graph* 24, 3 (2005), 965–972.